CSCE774 – Robotics
Fall 2011 – Homework 2

Assigned: September 27
Due: October 11, 12:30 pm

In this assignment, you will study nondeterministic information states by implementing some software. You may use any language and computing platform you prefer.

Problem description

Consider a system with the following elements.

- The state space is the entire plane, $X = \mathbb{R}^2$. There are no obstacles, nor any outer boundary.
- The action space is $U = [-5, +5] \times [-5, +5]$. (This is a Cartesian product of two closed intervals.)
- The nature action space is $\Theta = [-1/4, +3/4] \times [-3/4, +1/4]$.
- The state transition function is $f(x_k, u_k, \theta_k) = x + u + \theta$.
- The observation space is $Y = \{0, 1\} \times \mathbb{R}$.
- The nature observation action space is $\Psi = \{0, 1\} \times [-1.5, +1.5]$.
- Let (x_1, x_2) denote a state and $(\psi_{axis}, \psi_{offset})$ denote a nature observation action. The observation function is

  $$h((x_1, x_2), (\psi_{axis}, \psi_{offset})) = \begin{cases} (0, x_1 + \psi_{offset}) & \text{if } \psi_{axis} = 0 \\ (1, x_2 + \psi_{offset}) & \text{if } \psi_{axis} = 1 \end{cases}$$

  The intuition is that the robot senses *either* its horizontal position (when $\psi_{axis} = 0$) or its vertical position (when $\psi_{axis} = 0$). In addition, these measurements are perturbed by ψ_{offset}.

Tasks

Before implementing anything, study the problem statement and answer the following questions.

1. Suppose the robot receives the observation $y_k = (0, b)$. Describe the sensor preimage $H(y_k)$ of this observation. How does the sensor preimage change if the observation is $y_k = (1, b)$?
2. Suppose the robot is at state $x = (x_1, x_2)$ and executes action $u = (u_1, u_2)$. Describe the forward projection $F(x, u)$.
3. Suppose the robot starts at a known initial state. Describe the nondeterministic I-states that can be reached, across any number of stages, from this initial condition. (Hint: The correct answer can be formed by inserting a single word into the blank in the following sentence: "The nondeterministic I-state will always be a ________.")

Your answers to the previous three questions should provide some insight into how to represent and update the nondeterministic I-states for this system. Create software that enables you to complete the following tasks. For each task, start with the initial condition $\eta_0 = [-10, 10] \times [-10, 10]$.

4. Show each of the nondeterministic I-states that occur as the robot undergoes the following sequence of observations and actions:

(a) $y_1 = (0, 1/2)$

(b) $u_1 = (2, 2)$

(c) $y_2 = (1, 6)$

(d) $u_2 = (2, 2)$

5. Perform a simulation spanning 500 stages, in which both the robot and nature make their decisions at random. The initial state (unknown to the robot) is $(0, 0)$. In each stage k, keep track of x_k, u_k, θ_k, ψ_k, y_k, and $\kappa_{ndet}(\eta_k)$. Compute the area of the I-state after each observation update, and plot this area as a function of the stage index. What was the largest nondeterministic I-state, measured by its area, that occurred in this simulation? What was the smallest? Give a geometric description of these largest and smallest I-states, not just their areas.

6. Suppose that the robot's goal it to ensure that it reaches a point close to the origin. Specifically, the robot's goal region is a subset of the nondeterministic I-space:

$$\mathcal{I}_G = \{\eta_k \in \mathcal{I}_{ndet} \mid \eta_k \subseteq [-2, +2] \times [-2, +2]\}$$

Reaching this goal ensures that the robot is within two units in each direction of the origin.

   (a) Use your intuition to **design** a policy mapping nondeterministic I-states to actions, attempting to achieve this goal. There is room for creativity here, and strict optimality is not expected. However, you should ensure that your policy generates a clearly defined action for every reachable nondeterministic I-state. Carefully describe your policy in English and/or mathematical symbols.

   (b) **Implement** your policy, replacing the random actions from the previous task.

   (c) **Simulate** the robot executing your policy. Nature chooses θ ad ψ randomly (but the robot, not knowing this, still uses worst-case reasoning). Instead of executing a pre-defined number of stages, stop when the robot achieves its goal. Perform this simulation 1,000 times. What is the average number of stages required to achieve the goal? The fewest stages? The most stages?

Submitting your solution

Submit in class as hardcopy:

- A report answering the questions raised above.

Send by email:

- Any source code you wrote to complete the assignment.